

# A Fingerprint Method for Variability and Robustness Analysis of Stochastically Controlled Cellular Actuator Arrays

David L. MacNair and Jun Ueda

**Abstract**—This paper presents a “Fingerprint Method” for modeling and subsequently characterizing stochastically controlled actuator arrays. The actuator arrays are built from small actuator cells with structural elasticity. These cells are controlled using a bistable stochastic process wherein all cells are given a common input probability (control) value which they use to determine whether to actuate or relax. Arranging the cells in different networks gives different actuator array properties, which must be found before the actuator arrays can be applied to manipulators. The fingerprint method is used to describe and automatically generate every possible stochastic actuator array topology for a given number of cells, and to calculate actuator array properties such as: travel, required actuator strength/displacement, force range, force variance, and robustness for any array topology. The properties of several illustrative examples are shown and a discussion covers the importance of the properties, and trends between actuator array layouts and their properties. Finally, results from a validation experiment using a stochastically controlled solenoid array are presented.

**keywords:** Stochastic Control, Force Variability, Cellular Control System, Muscle, Compliant Actuator, Actuator Array Topology

## I. INTRODUCTION

Generation of natural movements, or the movements created by biological systems including humans and animals, has been one of the biggest scientific questions discussed in physiology for decades. Based on visual perception of “biological motion” (Johansson, 1973; Troje, 2002) humans can easily distinguish between motions created by artificial systems such as robots and those of humans, and human natural movements are stereotypical (RD. Crowninshield, et al., 1981; Van Bolhuis and Gielen, 1999; Buchanan and Shreeve, 1996; Prilutsky, 2000; Dickinson et al., 2000; Nozaki et al., 2005). Excluding small individual differences, motions with the same objective, such as arm-reaching or walking, tend to be very similar among different people. Stereotypical adaptation processes are also observed in neurological patients (Davies, 2000; Krishnamoorthy et al., 2003; Reisman and Scholz, 2003). Together, these observations imply the existence of general rules for coordination of multiple muscles to generate natural movements.

David L. MacNair (corresponding author) and Jun Ueda are with George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, 801 Ferst Drive, Atlanta, GA 30332-0405. E-mail: david.macnair@gatech.edu, jun.ueda@me.gatech.edu

Many research efforts searching for these underlying rules of natural movement focus on the variability of motor system timing spike generation. This variability in motor commands results in variability of muscle forces (Harris and Wolpert, 1998; Churchland et al., 2006; Jones et al., 2002; Stein et al., 2005; Todorov, 2005; Todorov, 2002; Hamilton et al., 2004; Osu et al., 2004). Harris and Wolpert showed that the standard deviation (SD) of the commanded signal varies proportionally with the mean of the command signal (Harris and Wolpert, 1998). They argued that this “signal-dependent” noise plays a central role in motor control and that movements are organized to minimize the variance at the endpoint. When taking a point-to-point motion, for example, Harris and Wolpert argue that the optimal strategy is to reach the desired endpoint with the minimum error due to the noise in the motor signals. The trajectory, velocity, and acceleration for the motion are determined by solving this optimization problem. They discovered that the calculated trajectories according to this criterion are similar to the stereotypical human motions, or natural movements (Harris and Wolpert, 1998). Todorov et al. (Todorov, 2005) suggested that this optimization strategy is equivalent to  $\min(\sum(\sigma^k))$ , the optimality principle in muscle force generation (RD. Crowninshield, et al., 1981; Van Bolhuis and Gielen, 1999; Buchanan and Shreeve, 1996; Prilutsky, 2000); that is, a minimization of the sum of muscular stress,  $\sigma$ , raised to a power,  $k$ , which is subject to the force/torque constraints of a given task. Although physiologically-based cost functions with this structure have been used in biomechanics and physiology to predict redundant muscle forces, its physiological meaning was not clear. Simmons et al. has applied this concept to optimal control of a standard 2-DOF manipulator with non-redundant planar rotary joints (Simmons and Demiris, 2005), however signal-dependent noise was computed and merely added to the control signal to mimic a natural system, but this noise never exists in AC/DC rotary motors. This is the beginning of a bottom-up approach, but large gaps still remain. The work presented by Harris and Wolpert is intuitively understandable and a likely solution to the natural movement problem, but more effort is needed to reach fully conclusive results.

The contrast between the actuators used in biological systems (i.e., muscles) and robotic systems (e.g., rotary DC motors) is another possible cause of the difference between movements of humans and machines. Biological muscles are completely different from their electromagnetic counterparts, both in constitution and function. (Martini and Bartholomew,

2006; Jacob et al., 1982; Yamaguchi, 2001; Kostyukov, 1998; Hospod et al., 2007; Tyreman and Molloy, 2003; Stern et al., 1997). They are energy efficient, compact, light weight, naturally compliant, and silent. As new robotic actuator technologies are studied, one of the goals is to invent artificial muscle actuators with these properties. In recent years, great progress in the development of new actuators has been presented using, e.g., shape memory alloys (Srinivasan and McFarland, 2001; Fu et al., 2004), pneumatic rubber actuators (Sanchez et al., 1998; Caldwell and Tsagarakis, 2002), conductive polymers (Hara et al., 2004; Plante and Dubowsky, 2006; Shahinpoor et al., 2000), and piezoelectric materials (Uchino, 1997; Conway et al., 2007; Niezrecki et al., 2001; Canfield and Frecker, 2000; Dogan et al., 1997; Dogan et al., 1994; Haertling, 1994; Janker et al., 1999; Newnham et al., 1993; Onitsuka et al., 1995). These novel actuators are particularly useful in human assistive technologies (Veneman et al., 2007; Alexander et al., 1992; Krebs et al., ; Lee and Sankai, 2002; Perry and Rosen, 2006; Toth et al., ; Tsagarakis and Caldwell, 2003; Ueda et al., 2008a; Ming et al., 2008) and biomedical robotic applications. Beyond the obvious engineering benefits, use of these new actuator technologies in robotic systems gives researchers an experimental platform to explore the basis behind human motion and robotic duplication of this motion.

It is known from prior literature that the activation of sarcomeres is not governed by deterministic control, but is instead affected by a stochastic process due to the diffusion of calcium ions (Kitamura and Yanagida, 2003). It is unknown to what extent this or other properties affect whether a movement is perceived to be natural, but developing artificial muscles with stochastic properties will hopefully shed more light on this active research area. The stochasticity in networks has been attracting increased attention from broad academic areas such as computer science (robust computer networks), biology (robustness of human immune systems), and urban engineering (robust transportation networks) (Shooman, 2002; Julius et al., 2008; Alon, 2006; Stelling et al., 2002; Nagurney and Qiang, 2007; Singer, 2006), all of which serve as references when designing and analyzing the stochastic systems. The stochastic actuator arrays proposed by the last author's group (Ueda et al., 2006; Ueda et al., 2007c; Ueda et al., 2008c; Ueda et al., 2008b; Ueda et al., 2007b; Ueda et al., 2007a; Odhner et al., 2006; Odhner and Asada, 2008) was selected as the example basis for the arrays in this paper due to its stochastic nature. The actuator arrays also have spring-like compliancy similar to that of organic muscle, so hopefully the two properties together provide an effective test bed for a bottom-up approach to study biological muscle and natural motion.

This paper presents a novel fingerprint method used describe and automatically generate every possible stochastic actuator array topology for a given number of cells, and to calculate actuator array properties such as: travel, required actuator strength/displacement, force range, force variance, and robustness for any array topology. (MacNair and Ueda, 2009). The proposed analysis is experimentally validated using a solenoid actuator array; however the same properties and calculations can be easily extended to systems using different actuator types such as pneumatic cylinders, piezoelectric actuators,

or other fast acting linear actuators so long as the same control method, cell definition, and cell structure are used. The ultimate goal of this project is to understand the mechanism of generating natural movements of skeletal mechanisms driven by the stochastically controlled actuator arrays presented in this paper and inspired by biological muscles.

## II. STOCHASTICALLY CONTROLLED CELLULAR ACTUATORS INSPIRED BY BIOLOGICAL MUSCLES

The detailed mechanism of biological muscles is still unknown; however, biological muscles suggest important design guidelines for new robot actuators. Muscles are dynamic systems with high degrees of internal freedom and relatively few inputs and outputs. A muscle is composed of numerous sarcomeres, small functional units which contract to provide varying levels of displacement and stiffness, and internal force, velocity, and displacement receptors, i.e. Golgi tendon organs and muscle spindles (Martini and Bartholomew, 2006; Jacob et al., 1982; Hospod et al., 2007; Tyreman and Molloy, 2003; Stern et al., 1997). In vertebrates, impulses from the brain are carried by the spinal cord to motor neurons which in turn distribute the impulses to the muscle fibers, via synaptic terminals at neuromuscular junctions, and activate the sarcomeres. It is known that the activation of sarcomeres is a stochastic process, rather than a deterministic control, because of the diffusion of calcium ions at these junctions (Kitamura and Yanagida, 2003). The brain does not know or control the displacement or force of any given sarcomer, only the overall muscle. This is part of the inspiration being using a stochastic process to control the actuator arrays. Due to the large redundancy of sarcomeres, Golgi tendon organs, muscle spindles, motor neurons, etc., even if some fraction of the components are not functional the rest of the functional components can compensate to maintain total functionality. This is critical for biological systems, and very useful when reproduced in robotic actuators.

The last author proposed a new architecture for robot actuators inspired by muscle properties, which in turn has a potential to be a novel approach to synthesize biologically-inspired actuators (Ueda et al., 2006; Ueda et al., 2007c; Ueda et al., 2008c; Ueda et al., 2008b; Ueda et al., 2007b; Ueda et al., 2007a). This concept, named "stochastic cellular actuator," connects small PZT actuator units, shown in Fig. 1, in series or in parallel to compose a macro-size actuator array. A new structure called a "nested rhombus multi-layer mechanism" (Ueda et al., 2007c; Ueda et al., 2008c; Ueda et al., 2008b) was proposed to amplify the displacement of

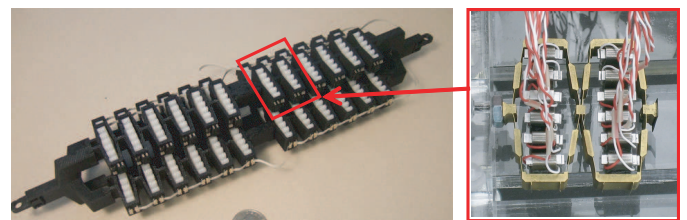


Fig. 1. Piezoelectric cellular actuator array

piezoelectric ceramic actuators, such as lead zirconate titanate (PZT), in order to create a novel actuator unit with 20-30% effective strain; which is comparable to natural skeletal muscles (Martini and Bartholomew, 2006; Jacob et al., 1982). This mechanism drastically mitigates the drawbacks of PZT, i.e. its extremely small strain of only 0.1 % (Uchino, 1997).

In addition to the strain amplification, the mechanisms are sufficient to produce “muscle-like” compliance in the actuator unit (Ueda et al., 2007c; Ueda et al., 2008c; Ueda et al., 2008b). These actuators are extremely compact and easy to use as building blocks in arrays for wide ranges of requirements in terms of force and displacement.

Figure 2 shows the concept of controlling a vast number of actuator units (Ueda et al., 2007b). Instead of wiring many control lines to each individual cell, each cellular actuator has a stochastic local control unit that receives a broadcasted signal from a central control unit and changes its state in a simple ON-OFF manner (Selden et al., 2006). This stochastic coordination, named “stochastic recruitment,” was inspired by the calcium diffusion process in the signal transduction of muscles (Kitamura and Yanagida, 2003) and drastically improves wiring and addressing issues. The ON-OFF control does not require high-performance actuator driver circuits and it resolves the problem associated with hysteresis of the actuator material. The stochastic cellular actuator array also has a high robustness against failure of the actuator units. Stochastic control theory (Kushner, 1965; Kushner, 1967; Doob, 1990) proves the system stability, and despite having no deterministic coordination, the ensemble of the cellular actuators robustly tracks a given trajectory even if, for example, 30% of the cells fail (Ueda et al., 2007b).

Although the characteristics of broadcast control have been investigated (Ueda et al., 2007b), this used only a simple array topology where actuator cells were connected in series and physical interaction among cells was neglected. This paper presents a method to generate and analyze complex actuator array topologies where actuator cells are connected in series, in parallel, or a mixture of both, interacting through mechanical compliance.

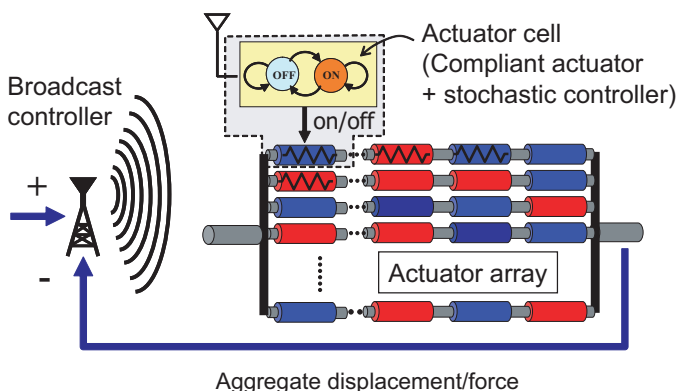


Fig. 2. Stochastic broadcast control of cellular actuator array

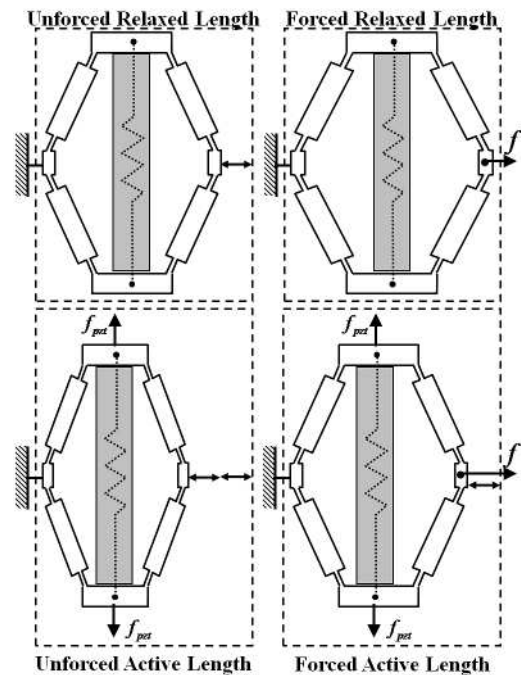


Fig. 3. Cell states and associated lengths

### III. MODELING AND CHARACTERIZING RECONFIGURABLE ACTUATOR ARRAY TOPOLOGIES

#### A. Modeling a single actuator array cell

For the purposes of this paper, a cell is a component of an actuator array consisting of a stack of piezoelectric actuators surrounded by a nested displacement amplification structure (Ueda et al., 2008b), as shown in Fig. 1. The piezoelectric actuator stack moves from a relaxed length to a shorter actuated length when activated and back to its relaxed length when deactivated. Assuming the external force acting on the cell does not overpower the actuator, the relaxed and actuated lengths of the actuator are constant. The amplification structure deforms according to the actuator displacement and forces applied externally to the cell, as shown in Fig. 3. As long as the amplification structure is kept in the linearly elastic portion of the stress strain curve, the structure acts as a linear spring. This is easily accomplished since the material amplification structure does not deform by 20% linearly, but rather the geometry of the structure allows for the amplified displacement. The bending which does occur is close enough to elastic bending of the high silicon bronze structure to be considered linear. Activating the piezoelectric actuator stack preloads the amplification structure (spring). If the cell sees no external force, this will cause the cell to shrink to the cell's unforced activate length. Alternatively, if the length of the cell is held constant, this will cause a force on the external constraint directly proportional to the change in length of the piezoelectric actuator stack. For this reason, each cell is modeled as a spring with a pure force generator acting in parallel. Figure 4 shows an equivalent system model for a piezoelectric actuator with an amplification structure. This model is applicable for any number of nesting levels (Ueda

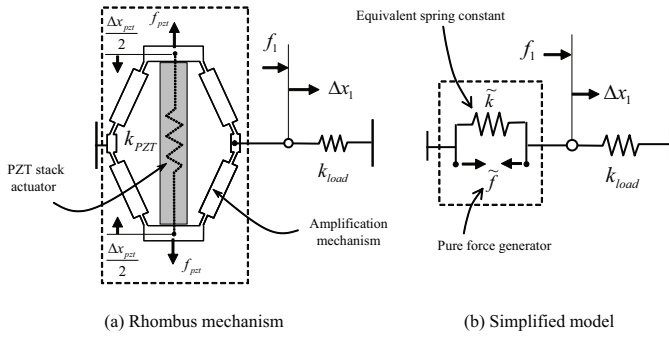


Fig. 4. Simplified representation of compliant actuator cell

et al., 2008b), but does not consider inertial or damping effects. Note that this paper uses piezoelectric actuators for examples, however the same modeling and techniques are applicable to compliant systems using different actuator types such as pneumatic cylinders, solenoids, or other fast acting linear actuators.

### B. Modeling an actuator array

Any one dimensional actuator array can be represented using four different element types, each having a certain number of variables, equations, and constants. Figure 5 shows an example topology. The element types are shown in Table I and Fig. 6, and are described below:

1) Node: A node ( $N_i$  for  $i = 1, 2, \dots, n$  where  $n$  is the number of nodes) is an imaginary unit used to connect the physical element types and to mathematically track the position and force along the actuator array. Node  $N_1$  and node  $N_n$ , the first and last nodes, are on the ends of the actuator array and only have one attached element each. These are used to represent the connection to the external environment. All other nodes are attached to two elements, one on the left and one on the right. Nodes have two variables. The position of  $N_i$

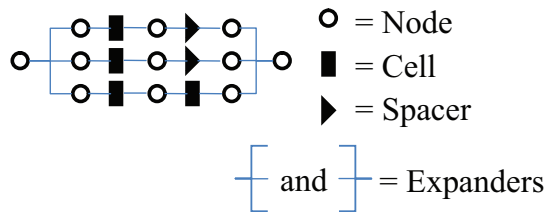


Fig. 5. Representation an actuator array topology

TABLE I  
ACTUATOR ARRAY ELEMENT TYPES

Type	Variable	Equations	Constants
Node $i$	Position $N_{i,x}$ Force $N_{i,f}$	—	—
Cell $j$	Disp. $d_j$	(1) Eqs. (2) (3)	Spring Constant $k_j$ Unforced Length $X_j$ Force Generator Force $F_j$
Spacer $y$	—	Eqs. (4),(4)	Length $q_y$
Expander	—	Eqs. (6),(7)	—

is  $N_{i,x}$  and the force of  $N_i$  is  $N_{i,f}$ . Nodes have no equations or given values.

2) Cell: A cell ( $C_j$  for  $j = 1, 2, \dots, J$  where  $J$  is the number of cells) has one variable, the displacement from the relaxed unforced length ( $d_j$ ), and three given values: the cell spring constant ( $k_j$ ), the cell relaxed unforced length ( $X_j$ ), and the pure force generator equivalent force ( $F_j$ ). Assuming all cells have the same force capability ( $F$ ),  $F_j = F$  when the cell is active and  $F_j = 0$  when the cell is relaxed. Cells also have three constituent equations. Equation (1) sets the position of the node  $N_{i+1}$  (node on the right) equal to the position of node  $N_i$  (node on the left) plus the combined distance of the relaxed unforced length ( $X_j$ ) and the displacement from the relaxed unforced length ( $d_j$ ). Equation (2) sets the force through node  $N_i$  equal to the cell spring constant ( $k_j$ ) times the displacement ( $d_j$ ) plus the pure force generator force ( $F_j$ ), and Equation (3) does the same for node  $N_{i+1}$ .

$$N_{i+1,x} - N_{i,x} - d_j = X_j \quad (1)$$

$$N_{i,f} - k_j \cdot d_j = F_j \quad (2)$$

$$N_{i+1,f} - k_j \cdot d_j = F_j \quad (3)$$

3) Spacer: A spacer ( $S_y$  for  $y = 1, 2, \dots, Y$  where  $Y$  is the number of spacers) is used to represent a constant length in the actuator array. They have no variables but have one given value, the length of the spacer ( $q_y$ ). Spacers also have two equations. Equation (4) sets the position of node  $N_{i+1}$  (node on the right) equal to the position of node  $N_i$  (node on the left) plus the length of the spacer ( $q_y$ ). Equation (5) sets the force through node  $N_{i+1}$  equal to the force through node  $N_i$ .

$$N_{i+1,x} - N_{i,x} = q_y \quad (4)$$

$$N_{i+1,f} - N_{i,f} = 0 \quad (5)$$

4) Expander: Expanders connect a single node on one side to multiple nodes on the other side without adding length to the system. This allows multiple cells to run in parallel, amplifying the force capacity and increasing redundancy of the actuator. Expanders have no variables or given values, but have  $h_e$  equations where  $h_e$  is the number of nodes connected

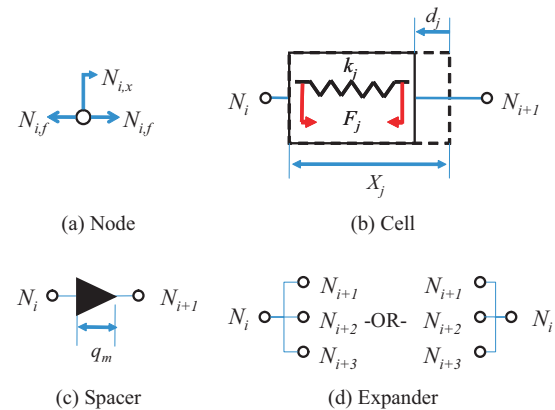


Fig. 6. Elements types used in creating actuator arrays



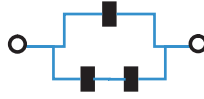


Fig. 7. A topology that cannot be represented with a fingerprint

by the expander.  $h_e - 1$  of the equations follow the form shown in (6) and make the positions of all of the connected nodes equal. The final equation, (7), adds the forces on the larger side of the expander and sets the result equal to the force on the smaller side. If a physical expander has length, a single spacer can be connected in series on the smaller side of the idealized expander.

$$\begin{cases} N_{i,x} - N_{i+1,x} = 0 \\ N_{i,x} - N_{i+2,x} = 0 \\ N_{i,x} - N_{i+3,x} = 0 \\ \vdots \end{cases} \quad (6)$$

$$N_{i,f} - N_{i+1,f} - N_{i+2,f} - N_{i+3,f} - \dots = 0 \quad (7)$$

Note that the equations list nodes sequentially about each element, but the node numbers will not necessarily be sequential. See Fig. 6 for what the node numbers correspond to for each element's description.

#### IV. FINGERPRINT REPRESENTATION

##### A. Fingerprint structures

Actuator arrays can be represented as lists of cells, spacers, and expanders with each element in the list knowing which nodes it connects to, but this method is bulky and makes describing and encoding actuator array topologies difficult. In order to more simply and compactly represent complex actuator arrays, a layer based description, or fingerprint, for actuator array topologies was developed.

The fingerprint assumes the actuator array can be separated into layers with each layer having the length of an unforced cell when the actuator array is fully relaxed and no external forces are applied. This means it can only describe systems which have no internal forces when all cells are relaxed and the array experiences no external force. For example, topologies shown in Fig. 7 cannot be represented. Following this assumption is generally a good design practice since internal forces can lead to buckling of actuator array segments, so the fingerprint method is considered robust enough to represent those systems most likely to be used in practice. An additional assumption holds that all cells are identical or that the data holding additional cell properties is stored elsewhere, however storage of this additional data is generally not difficult. This separates the geometry of the topology from the force properties of individual cells.

In order to represent an actuator array, it is first broken into layers of equal length as shown in Fig. 8. The layer break occurs just after the cells or spacers of the current layer and before the next set of expanders (or cells if no expanders exist in the next layer). Each layer can be represented using 2 parts, the front structure and the back structure, which share

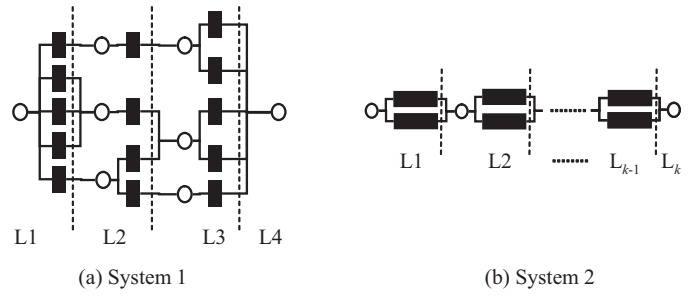


Fig. 8. Example array topologies

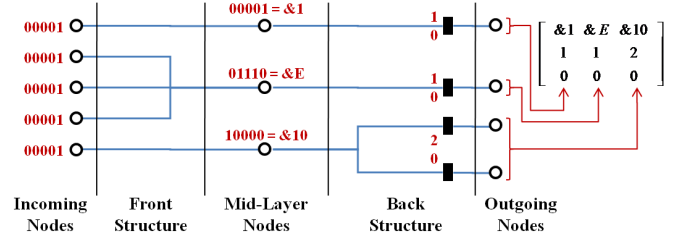


Fig. 9. System 1, layer 2 fingerprint description example

mid-layer nodes, as shown in Fig. 9. The front structure can be represented by  $h_m$  numbers, where  $h_m$  is the number of mid-layer nodes for layer  $m$ . Each front structure number is a hexadecimal representation of which incoming nodes connect to the corresponding mid-layer node. The first front structure number, for example, shows which incoming nodes connect directly to the first mid-layer node. In Fig. 9, the front structure would be represented by  $[\&1, \&E, \&10]$ . The representation can be decoded by converting the front structure numbers to binary.  $\&1$  corresponds to 00001 which means that the first mid-layer node connects only to the first incoming node.  $\&E$  corresponds to 01110 meaning the second mid-layer node connects to incoming nodes 2, 3, and 4.  $\&10$  corresponds to 10000 meaning the third mid-layer node connects to incoming node 5. Care must be taken to not mistakenly read the front structure number backwards, for instance mistaking 10000 as connecting to the incoming node one instead of incoming node five.

The back structure can be represented by 2 numbers per mid-layer node. These two numbers represent the number of cells and the number of spacers connected to that mid-layer node. Cells are always considered to connect to lower outgoing node numbers and spacers to higher outgoing node numbers when both cells and spacers connect to the same mid-layer node. This is discussed below. -1 for both the number of cells and the number of spacers is used to signify the end of the actuator array, which the build algorithm explicitly looks for to find the last layer. Having two 0's on any layer would represent a discontinuity, an error in the fingerprint. Having a 0 for the number of cells or number of spacers would not represent an error on any layer except the last since it is possible to have a layer with only cells or only spacers (only spacers would increase array length). The final fingerprint stacks the front structure numbers on top of the back structure numbers for mid-layer node in each layer. The final matrices for each

layer are  $3 \times h_m$  matrices which are appended to form the final fingerprint. For example, the array in Fig. 8 (a) has the following fingerprint:

$$\left[ \begin{array}{c|c|c|c|c|c|c|c} \&1 & \&1 & \&E & \&10 & \&1 & \&6 & \&8 & \&1F \\ 5 & 1 & 1 & 2 & 2 & 2 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{array} \right] \quad (8)$$

Similarly, the array in Fig. 8 (b) has

$$\left[ \begin{array}{c|c|c|c|c|c} \&1 & \&3 & \dots & \&3 & \&3 \\ 2 & 2 & \dots & 2 & -1 \\ 1 & 1 & \dots & 1 & -1 \end{array} \right]. \quad (9)$$

### B. Outgoing structure convention

The convention of putting cells above (lower node number) spacers helps avoid confusion. System 1 of Fig. 10 is shown not following the convention while system 2 is an identical topology which does follow the convention. System 3 shows how system 1 would be misinterpreted when the convention is not followed. When interpreted according to the stated convention, an algorithm would put the lower cell in system 1, layer 2, in direct parallel with the upper cell when it wrote the incoming connection for layer 3 (as shown by system 3). System 2, which correctly follows the convention, would be written as:

$$\left[ \begin{array}{c|c|c|c|c|c} \&1 & \&1 & \&2 & \&5 & \&2 & \&3 \\ 2 & 1 & 1 & 1 & 1 & -1 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{array} \right] \quad (10)$$

### C. Fingerprint spot-check

Properties which are necessary for a correct fingerprint include:

- 1) The top row's numbers should occupy every incoming connection point, no more than the number of connection points, and each incoming connection points can only be used once. This can be shown by adding the binary numbers of the first row together. The result should be equal to  $2^{h_I} - 1$  where  $h_I$  is the number of incoming nodes. For example, Layer 2 Fig. 8 (a) had the incoming connection numbers  $[\&1, \&E, \&10]$ .

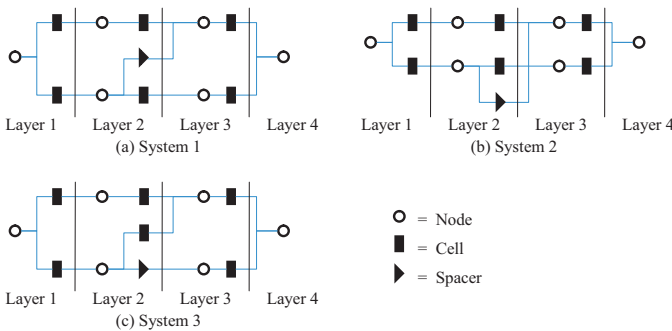


Fig. 10. Outgoing structure convention example

This translates to:  $00001 + 01110 + 10000 = 11111 = 2^5 - 1 = \&1F$ .

- 2) Sum of the last 2 rows of any layer equals the number of incoming points for the next layer and must be greater than zero.
- 3) The last layer must have -1's for the number of cells and the number of nodes.

### D. Fingerprint generation

In order to do an in-depth survey of actuator array properties, a method of generating all possible actuator array topologies given a fixed number of cells was needed. Since infinite topologies exist even given a finite number of cells, spacers were not included in the automatic generation of fingerprints. This makes the problem manageable, but does potentially remove some good topologies. Future work will address this issue, however the cell-only arrays provide valuable insight and spacers can always be added back manually. Analysis procedures shown later still apply when this constraint is relaxed.

The auto-generation problem was broken into a layer by layer iterative approach. Each layer was then further broken down into its front section, which contains the information for bringing the incoming nodes down to a certain number of mid-layer nodes, and its back section, which contains the information for expanding each mid-layer node to a certain number of cells and outgoing nodes, as was shown in Fig. 9.

For each front section, all of the possible methods for connecting the incoming nodes were explored using a recursive function. This function took in a binary input,  $a$ , showing which incoming nodes were not yet connected for the current test case. For example, in Fig. 11 incoming nodes 2 and 4 are not connected, thus the input would be 1010 which. The function iterates upward in binary from  $s = 1$  to  $s = 2^{L_a} - 1$  where  $L_a$  is the length of  $a$ . Each  $s$  has 0's concatenated to the left side until it is the same length as  $a$  and is then subtracted

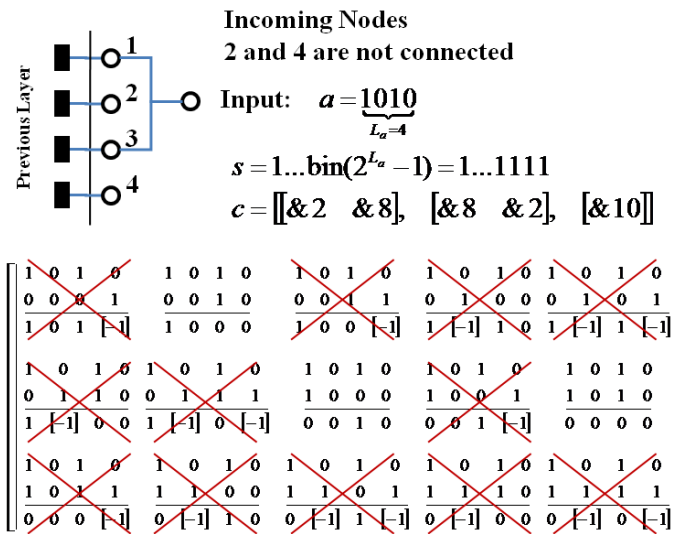


Fig. 11. Front section autogeneration example

from  $a$  element-wise. If there are any resulting -1's then  $s$  is tossed out, otherwise  $s$  is stored in hex in array  $c$  as  $c_l$ . This becomes front structure number as part of a fingerprint layer. To finish generating the front structure for each stored  $c_l$ , the function is called recursively with an input of  $a_n = a - c_l$ . No additional level of recursion is made for inputs of  $a_n = 0$ . The function returns the structure (11) if  $c$  has  $L$  entries with  $L > 1$ .

$$[c_1, \text{recursion}] ; \dots ; [c_l, \text{recursion}] ; \dots ; [c_L] \quad (11)$$

From the example illustrated in Fig. 11, the first entry in  $c$  is &2 representing a connection to only incoming node 2. The next layer of recursion then takes in an input of  $1010 - 0010 = 1000$  and returns &8. When the function finishes, the result is a structure of all of the frontal connection possibilities, each of which can have one or more mid-layer nodes. Since each mid-layer node must have at least one cells attached to it, those frontal connections which have more mid-layer nodes than remaining cells are tossed out.

Another recursive function is run to generate the back structure for each front structure. This function takes in the number of mid-layer nodes and the number of cells left to be placed, and returns all of the back connection possibilities. The function iterates from  $g = 1$  to  $g = G$  where  $G$  is given by  $G = z - m + 1$ ,  $z$  is the number of remaining cells, and  $m$  is the number of remaining mid-layer nodes. The number of cells connected to the first mid-layer node is equal to current value of iteration,  $g$ . The upper limit of the iteration is due to the need for each mid-layer node to have at least one cell attached. The result function is then recursively called with  $w$  remaining cells determined by  $w = z - g$  and  $o$  mid-layer nodes determined by  $o = m - 1$ .

Figure 12 shows a process tree for how the fingerprints were generated for actuators with 4 cells. All actuators begin with a single incoming node and a number of cells. The front end for layer 1 is always &1, and the back end uses between 1 and  $J$  cells where  $J$  is the total number of cells. The second and subsequent layers take each of the previous layer's configurations and generate all of the possible configurations for that layer. When the function is called with no remaining cells, the last layer is always filled in to connect all mid-layer nodes to a single output node and put -1's for the number of cells and the number of spacers.

Figure 13 shows the connection possibilities for 5 cells. The number of topologies for 2 – 8 cells are: 2 topologies for 2 cells, 4 topologies for 3 cells, 9 topologies for 4 cells, 23 topologies for 5 cells, 65 topologies for 6 cells, 199 topologies for 7 cells, 653 topologies for 8 cells, 2283 topologies for 9 cells, and 8467 topologies for 10 cells, respectively. For example, the computational time for 10 cells was 324 [s] by MATLAB running on a QuadCore 2.83GHz processor. The number of topologies and the computation time for generating all of the topologies increases exponentially with the number of cells.

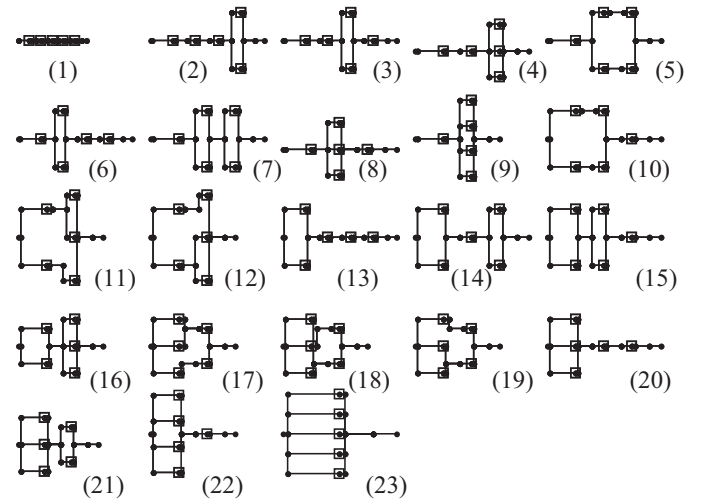


Fig. 13. Automatically generated 23 topologies for 5 cells

## V. ANALYSIS OF ACTUATOR ARRAY PROPERTIES

### A. Actuator array properties

**Number of Cells ( $J$ ):** As the number of cells increases, the actuator array cost increases, the power requirement increases, and the actuator array has a larger volume and mass. Increasing cells also generally increases the actuator array displacement and/or force capacity and decreases the normalized variance.

**Actuator Array Travel:** For the purposes of this paper, the actuator array displacement is considered to be half the difference between the relaxed unforced length and the active unforced length, i.e.,  $\Delta/2$  in Fig. 14. The relaxed unforced length is the length of the actuator array when all cells are relaxed and no external force is applied; this is also considered to be the minimum length. Similarly, the actuated unforced length is the length when all cells are active and no external force is applied. This length is shorter than the minimum length. The actuator array travel spans from the relaxed unforced length (minimum actuator array length) to this plus the displacement, as shown in Fig. 14. This assumption is explored further in the discussion about the use of actuator arrays in antagonistic pairs and ensures no compression forces exist across cells due to other cells activating.

**Force Function**  $\tau$  is a function of the probability input and the current actuator array length which yields the force an actuator array will provide. For a given length, a command of

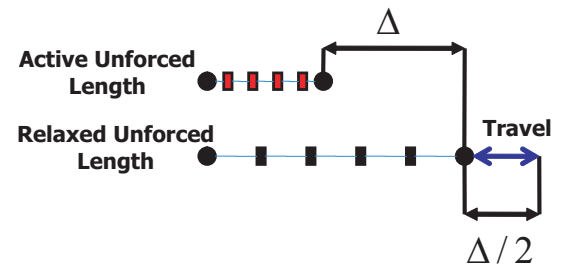


Fig. 14. Actuator array travel explanation

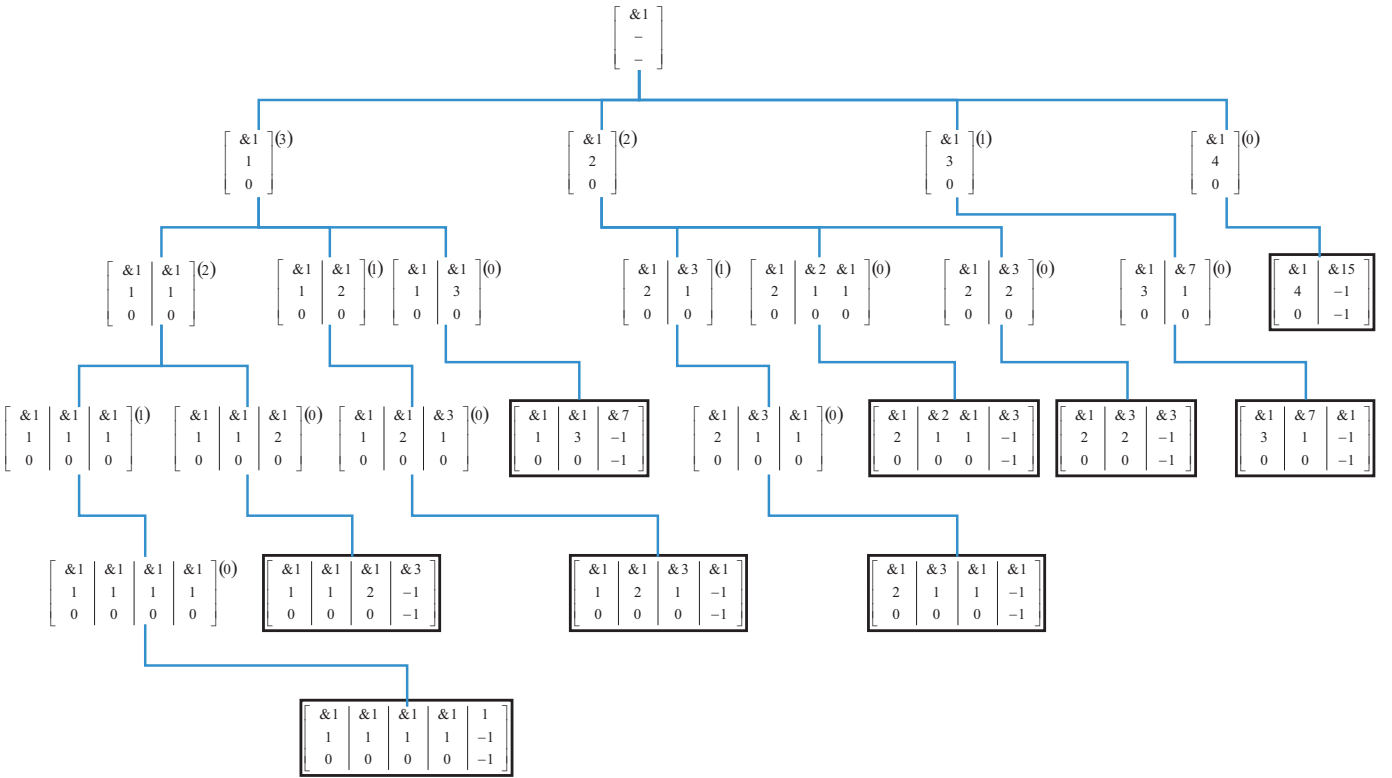


Fig. 12. Tree representing the automatic generation of a fingerprints for all 4 cell topologies

0% input probability will give the minimum possible force in the actuator array, a command of 100% will give the maximum possible force, and the command to mean-force relationship is linear between the two. A controller uses this function to achieve a desired mean force output.

**Required Actuator Force/Displacement (RAD / RAF)** Each piezoelectric or other type of linear actuator must have at least a certain force capacity, or required actuator force, in order to ensure it is able to actuate fully when loaded. Each actuator must also have a certain displacement which it will move to, but not beyond, whenever activated. For the purposes of this paper, it is assumed that there are no compression forces in the actuator array. This is a design restriction such that when the actuator array is in its relaxed unforced state no compression exists across any cell (considered good design practice), and is a result of the previously mentioned antagonistic pairs assumption.

**Variance Function** This is a function of the input probability value and the current actuator array length which yields the expected variance in the output force for an actuator array. While the mean force output will remain constant for a constant input probability value, the actual force output will vary over time. The larger the force variance, the farther from the mean value the force is likely to be at any point in time. Variance will lead to a greater potential for positioning error with open-loop control.

**Robustness (MCLU, WFFF)** The worst case failure of a cell is a break; meaning  $F_j$  and  $k_j$  for the cell are always zero, making the broken cell and all cells connected in series to the broken cell seem to vanish from the array. This is considered

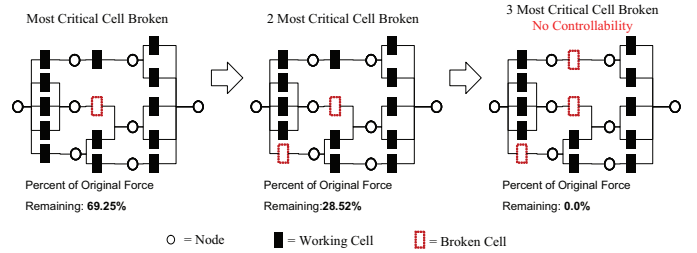


Fig. 15. Robustness measure: “minimum cell loss to uncontrollability”=3.

worse than simply having a non-functional cell since a non-functional cell still has an intact compliant structure and little to no affect on nearby cells. Two robustness measures were developed to characterize actuator arrays in terms of robustness. The “Minimum Cell Loss to Uncontrollability” (MCLU) defines, in the worst case scenario, how many cells would have to break to have zero controllable force capacity. The “Worst Failure Force Function” (WFFF) is the force function an actuator array can achieve after breaking a given number of its most critical cells (the cells which results in the lowest achievable total force once lost). These two measures are shown in Fig. 15. So long as forces on the actuator array are below this value, the actuator array will be able to function and have time to cope after the break of any cell.

### B. Relationship generation

To analyze the properties of the actuator arrays, the element equations described in Section III-B are arranged into three



systems of linear equations, or relationships, which are then used to solve for internal variables. Each relationship consists of an  $\mathbf{A}$  matrix containing the coefficients of the internal variables in the element equations, a  $\mathbf{B}$  vector containing the internal variables themselves, and a  $\mathbf{C}$  vector containing the given values (the right hand side of each equation). The system of linear equations is then solved using

$$\mathbf{B}_s = \mathbf{A}_s^{-1} \mathbf{C}_s \quad s = \{f, d, c\} \quad (12)$$

Depending on the type of analysis, these matrices contain slightly different elements. Matrices  $\mathbf{A}_f$ ,  $\mathbf{B}_f$ ,  $\mathbf{C}_f$  are for force relationship analysis,  $\mathbf{A}_c$ ,  $\mathbf{B}_c$ ,  $\mathbf{C}_c$  are for controllability analysis, and  $\mathbf{A}_d$ ,  $\mathbf{B}_d$ ,  $\mathbf{C}_d$  are for displacement relationship analysis.

1) *Force relationship*: The force relationship identifies all node forces given that certain cells are active and the actuator array has a given overall length ( $X_{tot}$ ). This relationship uses the element equations exactly as they appear in Table I but also adds (13) and (14), where node  $n$  is last node in the system. In the cell equations, the values of constant  $F_j$  are filled in as either  $F$  or 0 based on whether the cell is currently active or relaxed. Equation (12) solves for  $\mathbf{B}_f$ , the internal variables of the system. The force output for the actuator array is equal to the force in node 1, a component of vector  $\mathbf{B}_f$ .

$$N_{1,x} = 0 \quad (13)$$

$$N_{n,x} = X_{tot} \quad (14)$$

2) *Controllability relationship*: The controllability relationship is used to determine if an actuator array has any controllable force capacity. The relationship is the same as the force relationship except for two differences. First,  $F_j$  the pure-force generator force in the cell equation is moved to the left side of the equation as an unknown for all cells. This unknown force is assumed to be the same for all cells regardless of topology since the intent is to check for controllability, not to explicitly solve for individual cell forces. Second, (15) is added setting the force in node 1 and thus the actuator array output force equal to a test force of 1. Note that any non-zero test force is applicable. The test force of 1 is given for simplicity. If the  $\mathbf{A}_c$  matrix is full rank, the actuator array can change the exerted force by activating or relaxing cells. Figure 16 shows examples of an uncontrollable actuator arrays.

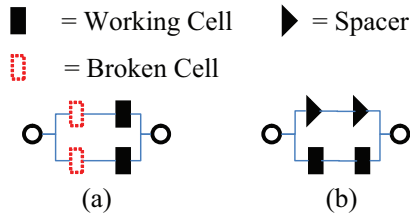


Fig. 16. Uncontrollable actuator arrays examples. In (a) all functional cells are negated by broken cells and unable to produce a force in the actuator array. In (b) spacers, constant length units, do not allow cell actuation to affect the length of the actuator array and internally cancel the forces created by the cells.

$$N_{1,f} = 1 \quad (15)$$

3) *Displacement relationship*: The displacement relationship is used to determine the unforced relaxed length and unforced actuated length of the actuator array. The relation is the same as the force relationship except that (13) is changed to (16) which sets the force in node 1 and thus the actuator array output force equal to 0. When run with all cells relaxed, this will give the unforced relaxed length and when run with all cells activated it will give the unforced activated length of the system.

$$N_{1,f} = 0 \quad (16)$$

### C. Example actuator topology representation

1) *Force relationship analysis*: An actuator array shown in Fig. 17 can be represented by (17). Assume that the both ends, represented by the first node ( $N_1$ ) and the last node ( $N_6$ ), are fixed and  $b_j = \{0, 1\}$  becomes 1 when the  $j$ -th cell is operational, and 0 when the cell is broken. These equations are further represented by a matrix-vector form as shown in (18).

$$\left. \begin{array}{l} N_{1,x} = 0 \\ N_{6,x} = X_{tot} \end{array} \right\} \text{Mount points}$$

$$\left. \begin{array}{l} N_{3,x} - N_{2,x} - d_1 = X_1 \\ N_{2,f} - b_1 \cdot k_1 \cdot d_1 = b_1 \cdot F_1 \\ N_{3,f} - b_1 \cdot k_1 \cdot d_1 = b_1 \cdot F_1 \\ N_{5,x} - N_{4,x} - d_2 = X_2 \\ N_{4,f} - b_2 \cdot k_2 \cdot d_2 = b_2 \cdot F_2 \\ N_{5,f} - b_2 \cdot k_2 \cdot d_2 = b_2 \cdot F_2 \end{array} \right\} \text{Cell Eqns.} \quad (17)$$

$$\left. \begin{array}{l} N_{1,x} - N_{2,x} = 0 \\ N_{1,x} - N_{4,x} = 0 \\ N_{1,f} - N_{2,f} - N_{4,f} = 0 \\ N_{6,x} - N_{3,x} = 0 \\ N_{6,x} - N_{5,x} = 0 \\ N_{6,f} - N_{3,f} - N_{5,f} = 0 \end{array} \right\} \text{Expander Eqns.}$$

2) *Displacement relationship analysis*: Assume that the end ( $N_1$ ) is fixed and the other end ( $N_6$ ) has a force  $F_{tot}$  applied, which can be zero. (19) gives the equations of the displacement relationship.

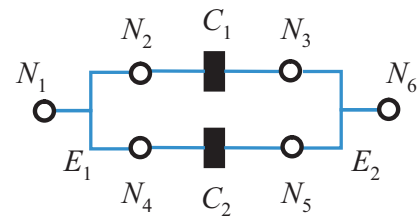


Fig. 17. Example array: two actuator cells are connected in parallel

$$\underbrace{\begin{bmatrix}
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -b_1 \cdot k_1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -b_1 \cdot k_1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -b_2 \cdot k_2 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -b_2 \cdot k_2 \\
 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0
 \end{bmatrix}}_{\mathbf{A}_f} \underbrace{\begin{bmatrix}
 N_{1,x} \\
 N_{1,f} \\
 N_{2,x} \\
 N_{2,f} \\
 N_{3,x} \\
 N_{3,f} \\
 N_{4,x} \\
 N_{4,f} \\
 N_{5,x} \\
 N_{5,f} \\
 N_{6,x} \\
 N_{6,f} \\
 d_1 \\
 d_2
 \end{bmatrix}}_{\mathbf{B}_f} = \underbrace{\begin{bmatrix}
 0 \\
 X_{tot} \\
 X_1 \\
 b_1 \cdot F_1 \\
 b_1 \cdot F_1 \\
 X_2 \\
 b_2 \cdot F_2 \\
 b_2 \cdot F_2 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0
 \end{bmatrix}}_{\mathbf{C}_f} \quad (18)$$

$$\left. \begin{array}{l}
 N_{1,x} = 0 \\
 N_{6,f} = F_{tot} \\
 N_{3,x} - N_{2,x} - d_1 = X_1 \\
 N_{2,f} - b_1 \cdot k_1 \cdot d_1 = b_1 \cdot F_1 \\
 N_{3,f} - b_1 \cdot k_1 \cdot d_1 = b_1 \cdot F_1 \\
 N_{5,x} - N_{4,x} - d_2 = X_2 \\
 N_{4,f} - b_2 \cdot k_2 \cdot d_2 = b_2 \cdot F_2 \\
 N_{5,f} - b_2 \cdot k_2 \cdot d_2 = b_2 \cdot F_2
 \end{array} \right\} \text{Cell Eqs.}$$

$$\left. \begin{array}{l}
 N_{1,x} - N_{2,x} = 0 \\
 N_{1,x} - N_{4,x} = 0 \\
 N_{1,f} - N_{2,f} - N_{4,f} = 0 \\
 N_{6,x} - N_{3,x} = 0 \\
 N_{6,x} - N_{5,x} = 0 \\
 N_{6,f} - N_{3,f} - N_{5,f} = 0
 \end{array} \right\} \text{Expander Eqs.}$$

(19)

$$\left. \begin{array}{l}
 N_{1,x} = 0 \\
 N_{6,x} = X_{tot} \\
 N_{3,x} - N_{2,x} - d_1 = X_1 \\
 N_{2,f} - b_1 \cdot k_1 \cdot d_1 - b_1 \cdot f_{com} = 0 \\
 N_{3,f} - b_1 \cdot k_1 \cdot d_1 - b_1 \cdot f_{com} = 0 \\
 N_{5,x} - N_{4,x} - d_2 = X_2 \\
 N_{4,f} - b_2 \cdot k_2 \cdot d_2 - b_2 \cdot f_{com} = 0 \\
 N_{5,f} - b_2 \cdot k_2 \cdot d_2 - b_2 \cdot f_{com} = 0
 \end{array} \right\} \text{Cell Eqs.}$$

$$\left. \begin{array}{l}
 N_{1,x} - N_{2,x} = 0 \\
 N_{1,x} - N_{4,x} = 0 \\
 N_{1,f} - N_{2,f} - N_{4,f} = 0 \\
 N_{6,x} - N_{3,x} = 0 \\
 N_{6,x} - N_{5,x} = 0 \\
 N_{6,f} - N_{3,f} - N_{5,f} = 0
 \end{array} \right\} \text{Expander Eqs.}$$

$$N_{1,f} - 1 = 0 \dots \text{Test force of 1 for Node 1}$$

(20)

Only the second equation changes from (18). By choosing  $\mathbf{B}_d$  as the same as  $\mathbf{B}_f$  in (18) and

$$\mathbf{C}_d = [0, F_{tot}, X_1, b_1 F_1, b_1 F_1, X_2, b_2 F_2, b_2 F_2, 0, 0, 0, 0, 0, 0]^T,$$

matrix-vector form  $\mathbf{A}_d$  can be obtained.

3) *Controllability analysis*: To investigate the robustness properties, the equations in (20), a modification of (17), are used. In ()  $f_{com}$  is a common force among all cells for a test force given to the node 1. This expands the original matrix form as shown in (21).

#### D. Property calculations

The actuator array travel ( $\Delta/2$ ) is the first property calculated for each actuator array. This is generated by using the displacement relationship to calculate the unforced relaxed length ( $N_{n,x}$  with all cells relaxed and  $F_{tot} = 0$ ) and unforced activated length ( $N_{n,x}$  with all cells activated and  $F_{tot} = 0$ ). As stated in section 5.1, the actuator array travel spans from the relaxed unforced length to this plus the displacement, as was shown in Fig. 14. The force function and force variance function are calculated using  $\mathbf{A}_f$ ,  $\mathbf{B}_f$ , and  $\mathbf{C}_f$  from the force relationship.  $\mathbf{C}_f$  can be separated into a vector containing the force components ( $\mathbf{G}$ ) and a vector containing all of the other components ( $\mathbf{H}$ ) as shown in (22). ( $\mathbf{G}$ ) contains two duplicate force entries for each cell, and each cell's pure-force can be modeled as an independent Bernoulli trial multiplied by the pure-force capacity of the cell ( $f_j$ ) as shown in (23), where  $r$  is the random value generated by the cell and  $p$  is the input probability as well as the expected value of the Bernoulli trial. The mean, or expected value, of  $\mathbf{B}_f$  is calculated using (24) (Hines et al., 2008).

$$\begin{array}{c}
\begin{array}{c} \mathbf{A}_f \end{array} \\
\begin{array}{c} \underbrace{\begin{array}{cccccccccccccccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}}_{\mathbf{A}_c} \end{array}
\end{array}
\begin{array}{c}
\begin{array}{c} 0 \\ 0 \\ 0 \\ -b_1 \\ -b_1 \\ 0 \\ -b_2 \\ -b_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \\
\begin{array}{c} \underbrace{\begin{array}{c} \mathbf{B}_f \\ f \\ \mathbf{B}_c \end{array}} = \end{array}
\end{array}
\begin{array}{c}
\begin{array}{c} 0 \\ X_{tot} \\ X_1 \\ 0 \\ 0 \\ X_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{array} \\
\begin{array}{c} \underbrace{\mathbf{C}_c} \end{array}
\end{array}
\quad (21)$$

$$\mathbf{C}_f = \begin{bmatrix} \mathbf{C}_f \\ c_1 \\ \vdots \\ c_i \\ F_j \\ F_j \\ c_{i+3} \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} \mathbf{G} \\ 0 \\ \vdots \\ 0 \\ F_j \\ F_j \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{H} \\ c_1 \\ \vdots \\ c_i \\ 0 \\ 0 \\ c_{i+3} \\ \vdots \\ c_n \end{bmatrix} \quad (22)$$

$$F_j = \begin{cases} 1 \cdot f_j & p > r_j \\ 0 & p \leq r_j \end{cases} \quad (23)$$

$$E[\mathbf{B}_f] = \mathbf{A}_f^{-1} \cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ f_j \cdot p \\ f_j \cdot p \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \mathbf{A}_f^{-1} \cdot \mathbf{H} \quad (24)$$

Since each of the Bernoulli trial is independent, the variance of  $\mathbf{B}$  can be calculated using (25):

$$Var[\mathbf{B}_f] = \mathbf{A}_f^{-1} \cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ f_j \cdot p \cdot (1-p) \\ f_j \cdot p \cdot (1-p) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (25)$$

The force function is  $E[N_{1,f}]$ , an entry in  $E[\mathbf{B}]$ . Likewise the variance function is  $Var[N_{1,f}]$ , an entry in  $Var[\mathbf{B}]$ . The force function is a linear function which shows that

the mean force will increase linearly with increasing input probability and/or actuator array length. The variance function is a quadratic function with roots at  $p = 0$  and  $p = 1$ , and a maximum at  $p = 0.5$ .

Given that a certain actuator array topology has already been chosen, a designer can find the Required Actuator Force/Displacement (RAF/RAD) for a desired force function ( $\tau$ ) by first finding a unit force function ( $\sigma$ ) for the actuator array given all  $f_j = 1, \forall j$ . A scaling factor ( $\nu$ ) can be found using (26).

$$\nu = \frac{\tau}{\sigma} \quad (26)$$

The scaling factor can then be plugged in for all  $\hat{f}_j$  in the force relationship. Solving the force relationship for  $\mathbf{B}_f$  will provide the forces in the nodes connected to each cell. This is the required actuator force,  $RAF_j$ , for each cell since it is the force the actuator driving the cell must be able to achieve.  $RAF_j$  tends to be the same as the pure-force generator force,  $F_j$  for all cells only in uniform topologies, such as with strictly parallel or strictly serial actuator arrays.  $RAF_j$  may need to be scaled for certain cell designs to match the amplification of the spring structure. For example, if an amplification structure decreases the force of a piezoelectric actuator by 20 times, the actual  $RAF_j$  will be 20 times higher than the  $RAF_j$  calculated using this method.

This method of calculating the required actuator force assumes all cells will have the same force capacity requirement. Designers of actuator arrays may decide to have a different distribution of force capacities,  $f_j$ 's, in an array. If this is the case, the distribution should be used instead of uniformly assigning all  $f_j = 1$  ( $j = 1, \dots, J$ ). Choosing this distribution is an optimization problem and is outside the scope of this paper. Once the scaling factor is found,  $\hat{f}_j$  can be found using

$$\hat{f}_j = f_j \times \nu \quad (27)$$

and is used in the force relationship to determine  $RAF_j$ .

The Required Actuator Displacement (RAD) for each cell is calculated using (28):

$$RAD_j = \frac{\hat{f}_j}{k_j} \quad (28)$$

This may vary from cell to cell depending on the cell spring constant, but it is the distance that must be achieved by the cell's actuator when activated to get the correct pure-force in the cell. A displacement greater or less than the RAD will affect the analysis and control of the actuator array. In some cases, it may be easier to choose an appropriate spring constant given  $\hat{f}_j$  and a fixed displacement,  $RAD_j$ . In this case, (28) would be solved for  $k_j$  instead of  $RAD_j$ .

The minimum cell loss to uncontrollability (MCLU) and the worst failure force function (WFFF) are calculated using a combination of the controllability relationship and the force relationship. First the system is input into the force relationship with all cells active and with the actuator array at its minimum travel. The most critical cell is determined as the cell carrying the highest force. This cell is then broken and the resulting actuator array is checked for controllability using the controllability relationship. If the actuator array still has a controllable force, it is once again fed into the force relationship with all cells active to determine the WFFF. This process is repeated until the actuator array is uncontrollable. The number of breaks required to make the actuator array uncontrollable is the Minimum Cell Loss to Uncontrollability.

## VI. NUMERICAL EXAMPLES

Figure 18 shows five actuator arrays which are analyzed; the results are shown below. In the analysis, all cells have a spring constant of  $k_j = 1$  for all  $j = 1, \dots, J$ . The force function,  $\tau$ , was selected such that the maximum output force was 1 when the actuator array was not stretched. The cells making up each topology are identical; however the RAD, RAF, and displacement vary between topologies. Figure 19 shows the force probability density functions (PDFs) for actuator array D. This graph shows the different forces which can be immediately achieved along with the probability that each point is reached given an input probability. Figure 20 shows a graphical representation of the force function, which is identical for all cases by choice. This figure shows how the mean force changes linearly with respect to input probability and displacement. Figure 21 shows the force variance curves for all of the actuator arrays. Interestingly, the same variance curves were obtained for arrays A, B, and C. Increasing the number of cells in an actuator topology generally results in a lower variance, as in E. Additionally, when cells do not uniformly carry the internal forces, as observed for D, they have a higher variance. The largest variance for any actuator array is obtained when the command signal, or input probability, is 0.5. This can be seen in the examples and is understandable since all of the actuator units will undoubtedly turn on if an input probability of 1 is given and off if an input probability of 0 is given.

Table II shows the number of cells, actuator array travel, required actuator force/displacement (RAD/RAF), minimum

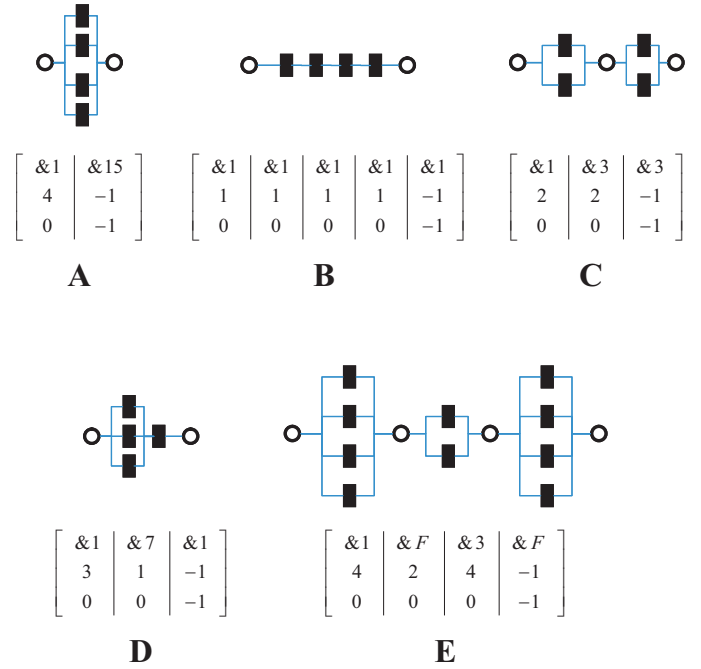


Fig. 18. Example actuator arrays analyzed using the fingerprint method

cell loss to uncontrollability (MCLU), and the percentage reduction between the original force function and the worst failure force function (WFFF) after breaking the most critical cell. A had the lowest RAD and RAF while B had the highest. A also had the lowest displacement while B had the highest, showing the trade-off between strength and displacement. B had a much longer travel than the other systems due to each cell having an additive effect on the overall displacement and since each actuator had to have a greater displacement in order to get the desired force function,  $\tau$ . Array A had the greatest robustness from both the MCLU and WFFF measurements while C and E were tied as the second best. B and D were both tied for the worst case, despite D having a parallel structure. This shows how having a parallel structure can improve robustness, but does not necessarily do so. E was shown as a contrast to the 4 cell examples to show how additional cells can improve the variance curve, strength (or RAF), or displacement. It also shows how cases with fewer cells can have more desirable properties, such as strength (lower RAF) when E is compared to A or displacement when E is compared to B.

## VII. EXPERIMENTAL VALIDATION

### A. Solenoid actuator array

In order to validate the theories presented in this paper, a solenoid based actuator array was developed. The array consisted of eighteen Magnet-Schultz of America (MSA) S-06683 solenoids, in a custom designed housing. The housing and connections between the solenoids and springs were printed using a Dimension bst 768 rapid prototyper. The solenoids each acted against two Lee Spring Co. LC 026EE 14M springs set in parallel, giving the same preloading effect as discussed



TABLE II  
EXAMPLE ACTUATOR ARRAY PROPERTIES

Topology	A	B	C	D	E
Number of Cells	4	4	4	4	10
Actuator Array Travel	0.125	2.0	0.5	0.667	0.5
Required Actuator Displacement	0.250	1.000	0.500	0.667	0.333
Maximum Required Actuator Force	0.250	1.000	0.500	1.000	0.500
Minimum Cell Loss to Uncontrollability	4	1	2	1	2
% Reduction After Losing Most Critical Cell	25.0 %	100.0%	33.3%	100.0%	33.3%

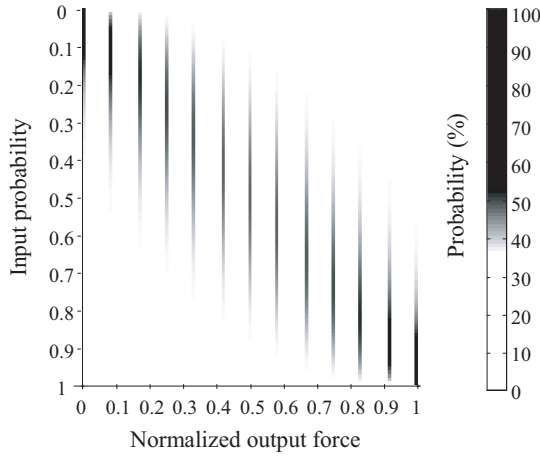


Fig. 19. Force probability density function for actuator array D.

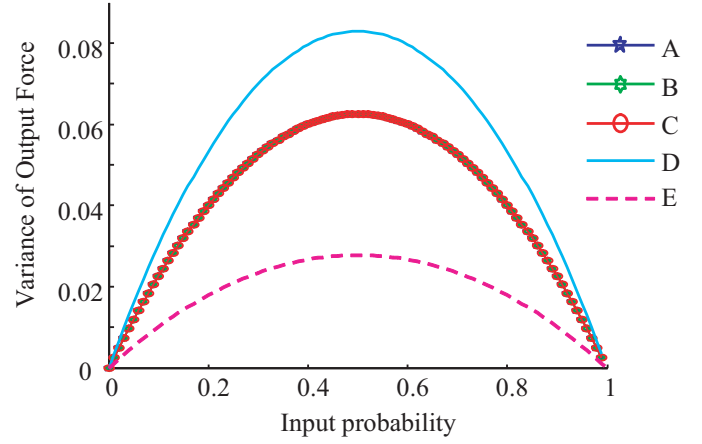


Fig. 21. Force variance of example arrays using normalized force function

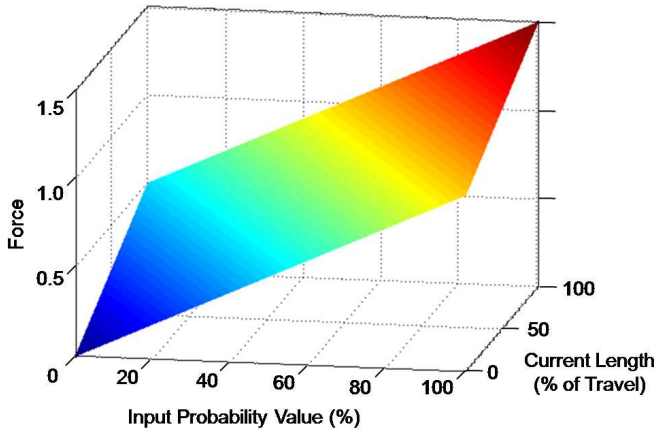


Fig. 20. Normalized force function for all array topologies

for a piezoelectric cell. Figure 22 (a) shows one solenoid cell. The cells were arranged into layers of three cells by three cells with all of the cells on each layer acting in parallel. Two of these layers were placed in series with one another giving the actuator array shown in Fig. 22 (b). Each cell had a combined spring constant of 438.7 N/m and maximum displacement of 2.5 cm which gave the effect of a 11.1 N pure force acting across the 438.7 N/m spring. Given the layout of the cells, this gave the actuator array a holding force capability of 200 N at its minimum length with an overall displacement capability of 5 cm.

The actuator array was controlled using LabView on a PC

as shown in Fig. 23. Due to limited output capability through the PC's parallel port, the cells were grouped according to Fig. 24 so that all eighteen cells could be controlled with only eight signal channels. This grouping also eliminated any moments which could have been generated due to an actuator firing on only one side of the actuator array. This was necessary to keep with the one-dimensional analysis presented in this paper. The fingerprint of this actuator array is represented by

$$\begin{bmatrix} \&1 & \&F & \&F \\ 4 & 4 & -1 \\ 0 & 0 & -1 \end{bmatrix}. \quad (29)$$

One of the cells on each layer consisted of three actuators while the others had two each. This detail could not be included in the fingerprint without encoding additional information. Since each actuator adds a small portion of the cells strength, the results of the experiment were averaged over ten runs, and only 2 of the eight cells were affected the overall results didn't not have a much greater variance than the model.

The LabView program took in an input probability value and, for each trial, generated eight random numbers. If the random number for a control channel was less than the input probability value, that control channel would actuate; otherwise it would relax. The solenoids provide a validation of the principles presented in this paper, but due to their high mass relative to their strength and their slow response time they were not suitable to test using a high frequency stochastic control scheme. Additionally, due to limitations in the solenoid's force capacity at maximum stroke length, each trial began with the actuator array at its unforced relaxed length. An input probability value was input and the cells

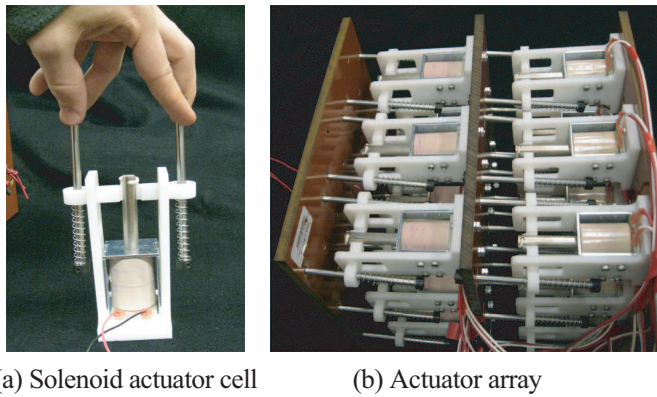


Fig. 22. Solenoid actuator array experimental setup

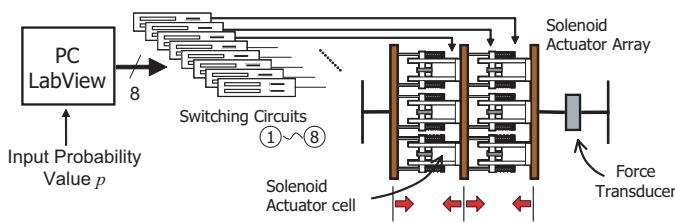


Fig. 23. Solenoid actuator array control diagram

actuated accordingly. The actuator array was then stretched to a preset length and the force required to achieve this stretch was measured using an OmegaDyne LCM703-50 force transducer. This allowed the experiment to test the actuator based on the holding force capacity of the solenoid as opposed to the significantly lower force at maximum stroke length.

### B. Force variance

Trials were run with input probability values ranging from zero to one with a step size of 0.1 giving a total of 110 trials. The minimum force required to achieve the stretch when no cells were active was subtracted from all data points and the data was normalized such that the maximum force required to achieve the stretch when all cells were active was 1. In Fig. 25(a) and (b), the normalized mean and variance of the forces were calculated from the results for each of the 11 input probability values and are shown along with the expected curve obtained by the analysis presented in this paper.

The experimental mean force and force variance curves closely matched the values calculated using the methods presented in this paper. Errors did exist in both but were expected due to the inherent error in using a probability driven system and due to inaccuracies in the force transducer. Had more trials been run at each of the input probability values, the experimental results would have more closely approached the theoretical values. The errors were low, and the results do validate the proposed analysis.

## VIII. DISCUSSION

The linearity and constant slope for each actuator array allows a controller to directly command a desired mean force

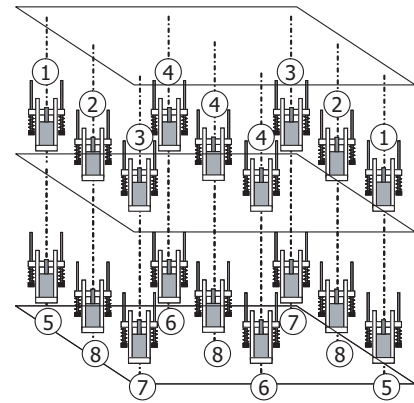


Fig. 24. Coupling among solenoid actuator array units. Units with the same number were treated as belonging to the same cell.

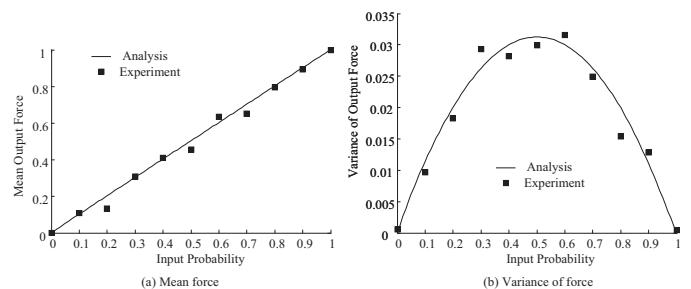


Fig. 25. Solenoid actuator array experimental results. (a) shows the mean force generated, which was highly linear. (b) shows the variance in the force which was close to the values calculated using the fingerprint method.

as long as the minimum and maximum forces at the endpoints of the actuator array's travel have been identified and the controller has knowledge of the actuator array's current length. Additionally, combining displacement and force sensors with the actuator array, the current belief of a manipulator and its payload can also be continuously updated by higher level controllers and learning techniques for more accurate manipulation. With these sensors, the actuator array can also be continuously calibrated so that it remains robust and accurate even despite multiple cell failures.

The variance of the muscle is a measure of the difference between the force commanded and the force delivered by an actuator array at any given point in time. Since the cells change position rapidly (200 Hz for piezoelectric actuator arrays), each instantaneous error provides only a small error in the overall impulse delivered and averages out quickly. Increasing the number of cells in an actuator array generally lowers the normalized variance; however variance also scales up with increasing force capacity. Cells which uniformly carry the internal forces generally have a lower variance. These observed properties are able to aid designers in creating arrays, however the reverse analysis is especially challenging due to the many design variables which must be manipulated to give any desired set of properties. The possible set of properties is also discrete (non-continuous) for any array when individual cell properties are pre-selected. For these reasons, this paper focuses on the forward analysis and leaves the reverse design

process to designers for their individual optimality criteria.

It should be noted that while one design criteria may be to minimize the variance, which can be made arbitrarily small by adding additional cells and adjusting the topology, arrays could be likewise constructed with larger variances to aid the development of theories regarding natural movement. This can provide evidence in support or against optimization techniques, such as minimization of jerk or minimization of control signal, contributing to research on the generation of natural movement and serving as a platform for future robots using natural movements.

The relationship between the input probability and variance shown in Figs. 21 and 25(b) is quadratic and does not monotonically increase, while the variance is expected to increase monotonically and proportionally with respect to the command input in biological systems (Harris and Wolpert, 1998). Maximum voluntary force may not be the maximum that a biological muscle can potentially generate; voluntary forces may be limited at the command level. The association between the input probability and variance could be approximated by a proportional function for a limited range, for example, for  $p = 0.0 \sim 0.4$ .

Future work will intensively investigate this difference between the artificial cellular actuator arrays and biological muscles in terms of variability. Note that Harris and Wolpert needed to introduce the "Size Principle" or an orderly requirement law in a top-down manner, assuming motor neurons that produce small forces are the first to be recruited and are followed by those that produce larger forces in order to recreate the proportional relationship (Jones et al., 2002). The artificial actuator systems, too, may need to introduce additional constraints in terms of the requirement order. One possibility is to take into account the force capacity degradation due to temperature when an actuator unit is exerting a constant force for a while.

Another future area of research concerns giving cells a force profile over time followed by a recharge period. This could more closely model biological muscle and would create the need for constraints on the range of the probability input, perhaps even limiting it to  $p = 0.0 \sim 0.4$  or less, since actuating all cells would be followed by a period of no force capacity despite a large probability value. It could also allow for higher force density actuators which act against a spring to provide large forces at the expense of requiring a recharge time to recoil the spring.

## IX. CONCLUSION

This paper presented a fingerprint method for modeling and subsequently characterizing stochastically controlled actuator arrays, which were inspired by biological muscle and the desire to create natural movements. The method automatically generates complex actuator array topologies and calculates the properties of the arrays such as travel, required actuator strength/displacement, force range, force variance, and robustness. The association between an input probability and force variance was investigated and two robustness criteria were introduced, the minimum cell loss to controllability failure

and the worst failure remaining force. Both of these measures become useful when an actuator array failure is critical to the application of the manipulator. The proposed analysis was validated by numerical examples and experiment.

## ACKNOWLEDGEMENT

This research was partially supported by NSF grant Cyber-Physical Systems ECCS-0932208.

## REFERENCES

- Alexander, M., Nelson, M., and Shah, A. (1992). Orthotics, adapted seating and assistive devices. *Pediatric Rehabilitation*, pages 186–187.
- Alon, U. (2006). *An introduction to systems biology: design principles of biological circuits*. CRC Press.
- Buchanan, T. and Shreeve, D. (1996). An evaluation of optimization techniques for the prediction of muscle activation patterns during isometric tasks. *Journal of biomechanical engineering*, 118:565.
- Caldwell, D. and Tsagarakis, N. (2002). Biomimetic actuators in prosthetic and rehabilitation applications. *Technology and Health Care*, 10(2):107–120.
- Canfield, S. and Frecker, M. (2000). Topology optimization of compliant mechanical amplifiers for piezoelectric actuators. *Structural and Multidisciplinary Optimization*, 20(4):269–279.
- Churchland, M., Afshar, A., and Shenoy, K. (2006). A central source of movement variability. *Neuron*, 52(6):1085–1096.
- Conway, N., Traina, Z., and Kim, S. (2007). A strain amplifying piezoelectric MEMS actuator. *Journal of Micromechanics and Microengineering*, 17(4):781–787.
- Davies, P. (2000). *Steps to Follow: The Comprehensive Treatment of Patients with Hemiplegia*. Springer.
- Dickinson, M., Farley, C., et al. (2000). How animals move: An integrative view. *Science*, 288(5463):100.
- Dogan, A., Uchino, K., and Newnham, R. (May 1997). Composite piezoelectric transducer with truncated conical end-caps "cymbal". *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 44(3):597–605.
- Dogan, A., Xu, Q., Onitsuka, K., Yoshikawa, S., Uchino, K., and Newnham, R. (1994). High displacement ceramic metal composite actuators (moonies). *Ferroelectrics*, 156(1):1–6.
- Doob, J. L. (1990). *Stochastic Processes (Wiley Classics Library)*. Wiley-Interscience, reprint edition.
- Fu, Y., Du, H., Huang, W., Zhang, S., and Hu, M. (2004). TiNi-based thin films in MEMS applications: a review. *Sensors & Actuators: A. Physical*, 112(2-3):395–408.
- Haertling, G. (1994). Rainbow Ceramics- A new type of ultra-high-displacement actuator. *American Ceramic Society Bulletin*, 73(1):93–96.
- Hamilton, A., Jones, K., and Wolpert, D. (2004). The scaling of motor noise with muscle strength and motor unit number in humans. *Exp Brain Res*, 157:417–430.
- Hara, S., Zama, T., Takashima, W., and Kaneto, K. (2004). Artificial muscles based on polypyrrole actuators with large

- strain and stress induced electrically. *Polymer journal*, 36(2):151–161.
- Harris, C. and Wolpert, D. (1998). Signal-dependent noise determines motor planning. *Nature*, 394(6695):780–784.
- Hines, W., Montgomery, D., and Borror, D. (2008). *Probability and statistics in engineering*. Wiley India Pvt. Ltd.
- Hospod, V., Aimonetti, J., Roll, J., and Ribot-Ciscar, E. (2007). Changes in human muscle spindle sensitivity during a proprioceptive attention task. *Journal of Neuroscience*, 27(19):5172.
- Jacob, S., Francone, C., and Lossow, W. (1982). *Structure and Function in Man*. WB Saunders Company.
- Janker, P., Christmann, M., Hermle, F., Lorkowski, T., and Storm, S. (1999). Mechatronics Using Piezoelectric Actuators. *Journal of the European Ceramics Society*, 19(6):1127–1131.
- Johansson, G. (1973). Visual perception of biological motion and a model for its analysis. *Perceiving events and objects*.
- Jones, K., Hamilton, A., and Wolpert, D. (2002). Sources of signal-dependent noise during isometric force production. *Journal of Neurophysiology*, 88(3):1533.
- Julius, A., Halasz, A., Sakar, M., Rubin, H., Kumar, V., and Pappas, G. (2008). Stochastic modeling and control of biological systems: the lactose regulation system of *Escherichia coli*. *IEEE Transactions on Automatic Control*, 53:51–65.
- Kitamura, K. and Yanagida, T. (2003). Stochastic properties of actomyosin motor. *Biosystems*, 71(1-2):101–110.
- Kostyukov, A. (1998). Muscle hysteresis and movement control: a theoretical study. *Neuroscience*, 83(1):303–320.
- Krebs, H., Celestino, J., Williams, D., Ferraro, M., Volpe, B., and Hogan, N. A wrist extension for MIT-MANUS. *Lecture notes in control and information sciences*, pages 377–390.
- Krishnamoorthy, V., Goodman, S., Zatsiorsky, V., and Latash, M. (2003). Muscle synergies during shifts of the center of pressure by standing persons: identification of muscle modes. *Biological Cybernetics*, 89(2):152–161.
- Kushner, H. (1965). On the Stability of Stochastic Dynamical Systems. *Proceedings of the National Academy of Sciences*, 53(1):8–12.
- Kushner, H. J. (1967). *Stochastic stability and control (Mathematics in science and engineering)*. Academic Press.
- Lee, S. and Sankai, Y. (2002). Power assist control for walking aid with HAL-3 based on EMG and impedance adjustment around knee joint. In *Intelligent Robots and System, 2002. IEEE/RSJ International Conference on*, volume 2.
- MacNair, D. and Ueda, J. (2009). Modeling and characterizing stochastic actuator arrays. In *the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- Martini, F. H. and Bartholomew, E. F. (2006). *Essentials of Anatomy & Physiology (4th Edition)*. Benjamin Cummings, 4 edition.
- Ming, D., Ueda, J., and Ogasawara, T. (2008). Pinpointed muscle force control using a power-assisting device: System configuration and experiment. *Biomedical Robotics and Biomechanics, 2008. BioRob 2008. 2nd IEEE RAS & EMBS International Conference on*, pages 181–186.
- Nagurney, A. and Qiang, Q. (2007). Robustness of transportation networks subject to degradable links. *Europhysics Letters*, 80(68001):1–6.
- Newnham, R., Dogan, A., Xu, Q., Onitsuka, K., Tressler, J., and Yoshikawa, S. (1993). Flextensional moonie actuators. In *1993 IEEE Proceedings. Ultrasonics Symposium*, volume 1, pages 509–513.
- Niezrecki, C., Brei, D., Balakrishnan, S., and Moskalik, A. (2001). Piezoelectric actuation: State of the art. *The Shock and Vibration Digest*, 33(4):269–280.
- Nozaki, D., Nakazawa, K., and Akai, M. (2005). Muscle Activity Determined by Cosine Tuning With a Nontrivial Preferred Direction During Isometric Force Exertion by Lower Limb. *Journal of Neurophysiology*, 93(5):2614–2624.
- Odhner, L. and Asada, H. H. (2008). Stochastic recruitment: A limited-feedback control policy for large ensemble systems. In *Robotics: Science and Systems IV*. MIT Press.
- Odhner, L., Ueda, J., and Asada, H. H. (2006). Feedback control of stochastic cellular actuators. In Khatib, O., Kumar, V., and Rus, D., editors, *ISER*, volume 39 of *Springer Tracts in Advanced Robotics*, pages 481–490. Springer.
- Onitsuka, K., Dogan, A., Tressler, J., Xu, Q., Yoshikawa, S., and Newnham, R. (1995). Metal-ceramic composite transducer, the “moonie”. *Journal of Intelligent Material Systems and Structures*, 6(4):447–455.
- Osu, R., Kamimura, N., Iwasaki, H., Nakano, E., Harris, C., Wada, Y., and Kawato, M. (2004). Optimal Impedance Control for Task Achievement in the Presence of Signal-Dependent Noise. *Journal of Neurophysiology*, 92(2):1199–1215.
- Perry, J. and Rosen, J. (2006). Design of a 7 Degree-of-Freedom Upper-Limb Powered Exoskeleton. In *Biomedical Robotics and Biomechanics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*, pages 805–810.
- Plante, J. and Dubowsky, S. (2006). On the nature of dielectric elastomer actuators and its implications for their design. In *Proceedings of SPIE*, volume 6168, page 61681J.
- Prilutsky, B. (2000). Coordination of two-and one-joint muscles: functional consequences and implications for motor control. *Motor Control*, 4(1):48–52.
- RD. Crowninshield, et al. (1981). A physiologically based criterion of muscle force prediction in locomotion. *J. Biomechanics*, 14:793–801.
- Reisman, D. and Scholz, J. (2003). Aspects of joint coordination are preserved during pointing in persons with post-stroke hemiparesis. *Brain*, 126(11):2510.
- Sanchez, A., Mahout, V., and Tondou, B. (1998). Nonlinear parametric identification of a McKibben artificial pneumatic muscle using flatness property of the system. In *Control Applications, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 1.
- Selden, B., Cho, K., and Asada, H. H. (2006). Segmented shape memory alloy actuators using hysteresis loop control. *Smart Materials and Structures*, 15(2):642–652.
- Shahinpoor, M., Kim, K., and Schreyer, H. (2000). Artificial



- sarcomere and muscle made with conductive polyacrylonitrile (C-PAN) fiber bundles. In *Proceedings of SPIE*, volume 3987, page 243.
- Shooman, M. (2002). *Reliability of computer systems and networks: fault tolerance, analysis and design*. Wiley-Interscience.
- Simmons, G. and Demiris, Y. (2005). Optimal robot arm control using the minimum variance model. *Journal of Robotic Systems*, 22(11):677–690.
- Singer, Y. (2006). Dynamic Measure of Network Robustness. In *2006 IEEE 24th Convention of Electrical and Electronics Engineers in Israel*, pages 366–370. Citeseer.
- Srinivasan, A. and McFarland, D. (2001). *Smart structures: analysis and design*. Cambridge Univ Pr.
- Stein, R., Gossen, E., and Jones, K. (2005). Neuronal variability: noise or part of the signal? *Nature Reviews Neuroscience*, 6(5):389–397.
- Stelling, J., Klamt, S., Bettenbrock, K., Schuster, S., and Gilles, E. (2002). Metabolic network structure determines key aspects of functionality and regulation. *Nature*, 420(6912):190–193.
- Stern, M., Pizarro, G., and Rios, E. (1997). Local control model of excitation-contraction coupling in skeletal muscle. *Journal of General Physiology*, 110(4):415.
- Todorov, E. (2002). Cosine tuning minimizes motor errors. *Neural Computation*, 14(6):1233–1260.
- Todorov, E. (2005). Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system. *Neural computation*, 17(5):1084–1108.
- Toth, A., Arz, G., Fazekas, G., Bratanov, D., and Zlatov, N. Post stroke shoulder-elbow physiotherapy with industrial robots. *Lecture notes in control and information sciences*, pages 391–411.
- Troje, N. (2002). Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of Vision*, 2(5):371–387.
- Tsagarakis, N. and Caldwell, D. (2003). Development and Control of a ?eSoft-Actuated?fExoskeleton for Use in Physiotherapy and Training. *Autonomous Robots*, 15(1):21–33.
- Tyreman, M. and Molloy, J. (2003). Molecular motors: nature’s nanomachines. In *IEE Proceedings-Nanobiotechnology*, volume 150, page 95.
- Uchino, K. (1997). *Piezoelectric Actuators and Ultrasonic Motors*. Kluwer Academic Publishers.
- Ueda, J., Matsugashita, M., Oya, R., and Ogasawara, T. (2008a). Control of muscle force during exercise using a musculoskeletal-exoskeletal integrated human model. In *Experimental Robotics*, pages 143–152.
- Ueda, J., Odhner, L., and Asada, H. H. (2007a). Broadcast feedback for stochastic cellular actuator systems consisting of nonuniform actuator units. In *Proceedings of 2007 IEEE International Conference on Robotics and Automation (ICRA ’07)*, pages 642–647.
- Ueda, J., Odhner, L., and Asada, H. H. (2007b). Broadcast Feedback of Stochastic Cellular Actuators Inspired by Biological Muscle Control. *The International Journal of Robotics Research*, 26(11-12):1251–1265.
- Ueda, J., Odhner, L., Kim, S.-G., and Asada, H. (2006). Distributed stochastic control of mems-pzt cellular actuators with broadcast feedback. In *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechanics (BioRob 2006)*, pages 272–277.
- Ueda, J., Secord, T., and Asada, H. (2007c). Design of PZT cellular actuators with power-law strain amplification. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007*, pages 1160–1165.
- Ueda, J., Secord, T., and Asada, H. (2008b). Piezoelectric Cellular Actuators Using Nested Rhombus Multilayer Mechanisms. In *1st Annual Dynamic Systems and Control Conference (DSCC 2008)*.
- Ueda, J., Secord, T., and Asada, H. (2008c). Static lumped parameter model for nested PZT cellular actuators with exponential strain amplification mechanisms. In *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008*, pages 3582–3587.
- Van Bolhuis, B. and Gielen, C. (1999). A comparison of models explaining muscle activation patterns for isometric contractions. *Biological cybernetics*, 81(3):249–261.
- Veneman, J., Kruidhof, R., Hekman, E., Ekkelenkamp, R., Van Asseldonk, E., and van der Kooij, H. (2007). Design and Evaluation of the LOPES Exoskeleton Robot for Interactive Gait Rehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 15(3):379.
- Yamaguchi, G. (2001). *Dynamic modeling of musculoskeletal motion*. Kluwer Academic Publishers.